The (2**(n+1)-2)-ray algorithm

Talman, A.J.J.; Doup, T.M.; van der Laan, G.

Published in:
Mathematical Programming

Publication date:
1987

Mathematical Programming 39 (1987) 241-252
North-Holland

THE $(2^{n+1}-2)$-RAY ALGORITHM: A NEW SIMPLICIAL ALGORITHM TO COMPUTE ECONOMIC EQUILIBRIA

T.M. DOUP

Department of Econometrics, Tilburg University, Tilburg, The Netherlands

G. van der LAAN

Department of Actuarial Sciences and Econometrics, Free University, Amsterdam, The Netherlands

A.J.J. TALMAN

Department of Econometrics, Tilburg University, Tilburg, The Netherlands

Received 25 June 1985
Revised manuscript received 4 November 1986

A new variable dimension simplicial restart algorithm is introduced to compute economic equilibria. The number of rays along which the algorithm can leave the starting point differs from the thusfar known algorithms. More precisely, the new algorithm has one ray to each of the $2^{n+1}-2$ faces of the n-dimensional price simplex, whereas the existing algorithms have $n+1$ rays either to each facet or to each vertex of the unit simplex. The path of points followed by the algorithm can be interpreted as a globally and universally convergent price adjustment process. The process is also economically meaningful and therefore it is a good alternative for the well-known Walras' tatonnement process. Computational results show that the algorithm is competitive with the most efficient simplicial algorithms developed thusfar.

Key words: Excess demand, equilibrium, simplicial algorithm, vector labelling.

1. Introduction

To find a zero point of a continuous excess demand function $z: S^n \to R^{n+1}$ where $S^n = \{x \in R^{n+1}_+ | \sum_j x_j = 1\}$ and z satisfies both $z_i(x) > 0$ when $x_i = 0$ and Walras' law $x^T z(x) = 0$ for all x, several simplicial algorithms have been developed (Scarf [12, 13], Kuhn [3, 4], Kuhn and MacKinnon [5], van der Laan and Talman [6, 7] and Doup and Talman [1]). In a simplicial subdivision or triangulation of S^n such algorithms search for a simplex which yields an approximate equilibrium by generating a path of adjacent simplices. The simplex with which the algorithm terminates is found within a finite number of steps. The so-called variable dimension algorithm, originally developed in van der Laan and Talman [6] can be started in an arbitrarily chosen grid point of the subdivision and generates a path of adjacent simplices of varying dimension. When the end simplex does not provide an approximate zero of z with sufficient accuracy, the subdivision is refined and the algorithm is restarted

This work is part of the *VF*-program "Equilibrium and Disequilibrium in Demand and Supply", which has been approved by the Netherlands Ministry of Education and Sciences.

in or close to the last found approximation. In general, the accuracy of an approximate equilibrium improves when the subdivision is refined.

By generating a sequence of simplices a piecewise linear path is traced which is determined by the piecewise linear approximation Z of z with respect to the underlying triangulation. More precisely, for any proper subset T of $I_{n+1} = \{1, \ldots, n+1\}$, let the $|T|$-dimensional convex subset $A(T)$ with $|T|$ the cardinality of T be defined by

$$A(T) = \{x \in S^n \mid x = v + \textstyle\sum_{j \in T} \lambda_j (e(j) - v), \lambda_j \geqq 0, j \in T\},$$

where v is the arbitrarily chosen starting point of the algorithm and where $e(j)$ is the j-th unit vector in R^{n+1}, $j \in I_{n+1}$. Clearly $A(T)$ is a $|T|$-dimensional cone in S^n with apex v. There are $n+1$ one-dimensional cones $A(\{i\})$, $i = 1, \ldots, n+1$, which are the rays of the algorithm. Each $|T|$-dimensional cone $A(T)$ is subdivided by the V-triangulation of S^n in $|T|$-dimensional simplices of. A detailed description can be found in Doup and Talman [1]. Now the path traced by the variable dimension algorithm is as follows. At the starting point $x = v$, let i be the (unique) index for which $z_i(v) = \max_h z_h(v)$. Then the algorithm increases λ_i away from zero, i.e., the starting point v is left along the ray $A(\{i\})$. In this way the ith component of x is increased and all the other components of x are decreased proportionally until a point x in $A(\{i\})$ is reached for which $Z_j(x) = Z_i(x)$ for some $j \neq i$. Then the algorithm traces a piecewise linear path of points x in $A(\{i, j\})$ by increasing λ_j away from zero and keeping $Z_j(x) = Z_i(x) = \max_h Z_h(x)$. In general the algorithm traces for varying $T \subset I_{n+1}$ a piecewise linear path in $A(T)$ such that a point x on the path satisfies

$$Z_i(x) = \max_h Z_h(x) \quad \text{if } i \in T,$$

and (1.1)

$$Z_i(x) \leqq \max_h Z_h(x) \quad \text{if } i \notin T.$$

So a point x on this path in $A(T)$ satisfies the complementarity property

$$x_j \geqq b(x) v_j \quad \text{and} \quad Z_j(x) = \max_h Z_h(x) \ \text{ if } j \in T$$

and

$$x_j = b(x) v_j \quad \text{and} \quad Z_j(x) \leqq \max_h Z_h(x) \ \text{ if } j \notin T$$

where $0 \leqq b(x) = 1 - \sum_{i \in T} \lambda_i(x)$, with the nonnegative $\lambda_i(x)$'s uniquely determined by

$$x = v + \sum_{i \in T} \lambda_i(x)(e(i) - v).$$

When for some $k \notin T$ an inequality in (1.1) becomes an equality the index k enters T and the algorithm continues in $A(T \cup \{k\})$ by increasing λ_k away from zero. On the other hand, when for some $j \in T$, λ_j becomes zero for a point x on the path in

$A(T)$, then the index j is deleted from T and the algorithm continues in $A(T\backslash\{j\})$ by decreasing $Z_j(x)$ away from $\max_h Z_h(x)$. As soon as T becomes I_{n+1} we have $Z_i(x) = \max_h Z_h(x)$ for all $i \in I_{n+1}$. Since $x^T z(x) = 0$ it follows that an approximate zero of z is found. The algorithm can be restarted in or close to this point with a finer grid in order to improve the accuracy.

The variable dimension algorithm mentioned above has the property that the starting point v is left along one out of $n+1$ rays. Two other variable dimension algorithms with $n+1$ rays were given in van der Laan and Talman [6, 7]. In the first one the well-known Q-triangulation of S^n underlies the algorithm. Then an initial increase of the component of v for which the z-value is maximal is compensated by a decrease with the same amount of just one other component of v. Although the increase of the i-th component of v in case $z_i(v) = \max_h z_h(v)$ is obvious, the decrease of only one other component of v in order to keep the sum of the x_i's equal to one seems to be unnatural. In the algorithm of van der Laan and Talman [7] the so-called U-triangulation of the affine hull of S^n underlies the algorithm. In this case the point v is left along a ray on which again one component of v is increased, but along which all the other components of v are decreased with a same amount. In Zangwill and Garcia [14] and van der Laan and Talman [8] the path followed by the algorithm based on this triangulation is interpreted as a globally convergent adjustment process. When applied to the equilibrium price vector problem in an exchange economy this adjustment process could serve as a globally convergent alternative for the classical Walras' tatonnement process. In Scarf [11] examples of reasonable economies have been given for which the Walras' price adjustment process may fail to converge to an equilibrium price vector. In fact recently it has been shown in Saari [10] that there does not exist any globally convergent iterative adjustment process based on a finite amount of information.

In the classical Walras' tatonnement process the prices of all the commodities are continuously adjusted proportionally to the value of their excess demands. In the processes induced by the simplicial algorithms developed thusfar initially only the price of the commodity with the highest excess demand is increased, while the prices of some or all of the other commidities are decreased in order to keep the sum of the prices equal to one. In this paper we present a simplicial variable dimension restart algorithm on S^n which as in Walras' tatonnement process initially increases the prices of the commodities for which there is excess demand and decreases the prices of the commodities with excess supply. These adjustments are proportional to the initial value of the prices. Along the path traced by the algorithm the prices of the commodities with positive (negative) excess demand are all proportionally larger (smaller) than the initial prices, while the prices of the commodities in equilibrium lie between the relative bounds induced by these two proportions. From an economic viewpoint this behaviour is much more intuitive than for the other simplicial processes. Moreover, contrary to Walras' tatonnement process, the algorithm is globally convergent in the sense that barring degeneracy the algorithm converges for any starting point and any excess demand function.

The computational results with the new algorithm are competitive with those of the most efficient $(n+1)$-ray algorithm (Doup and Talman [1]). Since for the new algorithm the number of rays to leave the starting point is equal to $2^{n+1}-2$, we call it the $(2^{n+1}-2)$-ray algorithm on S^n.

The paper is organized as follows. In Section 2 we give the piecewise linear path of the new algorithm. Section 3 describes the underlying subdivision of S^n. Section 4 discusses the linear programming pivot steps and the replacement steps of the algorithm. Finally, Section 5 gives some computational experience.

2. The path of the algorithm

The algorithm leaves the starting point v along the ray in S^n on which the components of v with positive z-value are proportionally increased and those with negative z-value are proportionally decreased. In the particular case that only one component of $z(v)$ is positive the ray leads from v to a vertex (zero-dimensional face) of S^n. In fact there leads a ray from v to each face of S^n. Since a face of S^n is the convex hull of a proper subset of the $n+1$ vertices of S^n, there are $2^{n+1}-2$ faces in S^n and therefore there are $2^{n+1}-2$ rays. The sign pattern of $z(v)$ determines along which ray the algorithm leaves v. The algorithm moves along this ray until a point y is reached where $Z_h(y)$ is equal to zero for some h, $1 \leqslant h \leqslant n+1$. Then the algorithm continues from y along a piecewise linear path of points x for which $Z_h(x)$ is kept equal to zero while the components of x not equal to h are still increased (decreased) proportionally to the same components of v according whether these components of $Z(x)$ are positive (negative). In general the algorithm traces a piecewise linear path of points x in S^n satisfying for all j

$$\begin{aligned} x_j &= av_j \quad \text{if } Z_j(x) > 0, \\ bv_j \leqslant x_j &\leqslant av_j \quad \text{if } Z_j(x) = 0, \\ bv_j &= x_j \quad \text{if } Z_j(x) < 0, \end{aligned} \tag{2.1}$$

with $0 < b \leqslant 1 \leqslant a$. Notice that $Z_j(x)$ is always positive if $x_j = 0$, so that the path cannot cross the boundary of S^n, i.e., b cannot become equal to zero. When comparing a point x on the path with the starting point v we have that all components x_j for which the piecewise linear excess demand $Z_j(x)$ is positive (negative) are the same factor a (b) larger (smaller) than v_j, while the components x_j with zero excess demand lie between bv_j and av_j. So, for each index j either $Z_j(x) = 0$ or x_j is equal to one of the two relative bounds bv_j or av_j depending on whether $Z_j(x)$ is negative or positive. The algorithm terminates as soon as a point x^* is reached for which either $Z(x^*) \leqslant 0$ or $Z(x^*) \geqslant 0$. Since for all j, x_j^* cannot be equal to zero and because of Walras' law, such a point x^* is an approximate solution to the equilibrium problem.

The existence of the piecewise linear path of points satisfying (2.1) from v to an approximate equilibrium can be proved by introducing a primal-dual pair L of subdivided manifolds and a function from L to R^{n+1} whose zero points satisfy (2.1), see e.g. Kojima and Yamamoto [2]. Under some nondegeneracy assumption, see also assumption 4.2, there is a piecewise linear path of zero points of this function in L connecting the pair $(v, z(v))$ with a point $(x^*, Z(x^*))$, where either $Z(x^*) \leqq 0$ or $Z(x^*) \geqq 0$. The projection of this path on S^n is the piecewise linear path of the algorithm. For a detailed description of the existence proof we refer to van der Laan and Talman [9]. In this paper we are mainly concerned with the computational procedure.

The piecewise linear path of points from v which satisfies (2.1) is followed by the algorithm through alternating replacement steps in the V-triangulation of S^n and pivot steps in a linear system of equations. These steps are described in Section 4. The next section describes how the V-triangulation underlies the algorithm.

3. The subdivision of S^n

To describe the subsets of S^n in which the new simplicial variable dimension algorithm operates, let s be an arbitrary sign vector in R^{n+1} with components $s_j \in \{-1, 0, +1\}$. Furthermore, let

$$I^-(s) = \{i \in I_{n+1} \mid s_i = -1\},$$

$$I^0(s) = \{i \in I_{n+1} \mid s_i = 0\},$$

$$I^+(s) = \{i \in I_{n+1} \mid s_i = +1\}.$$

In the sequel we assume that both $|I^+(s)|$ and $|I^-(s)|$ are at least equal to one, so that at least one component of s is equal to $+1$ and at least one component of s is equal to -1. Observe that there are $2^{n+1}-2$ of such sign vectors containing no zeroes at all. Each sign vector s will induce a t-dimensional subset $A(s)$ of S^n with $t = |I^0(s)| + 1$. Notice that t lies between 1 and n and is equal to one for the $2^{n+1}-2$ sign vectors containing no zeroes at all. Finally, let v be the starting point of the algorithm. We assume that v lies in the interior of S^n.

Definition 3.1. Let s be a sign vector with $|I^+(s)|$ and $|I^-(s)|$ positive. Then

$$A(s) = \{x \in S^n \mid x_i = av_i \text{ if } i \in I^+(s),\ bv_i \leqq x_i \leqq av_i \text{ if } i \in I^0(s), \text{ and}$$

$$x_i = bv_i \text{ if } i \in I^-(s), \text{ with } 0 \leqq b \leqq 1 \leqq a\}.$$

The boundary of a t-dimensional $A(s)$ consists of the $(t-1)$-dimensional sets $A(s')$ with $s'_i = \pm 1$ for exactly one i in $I^0(s)$ and $s'_h = s_h$, $h \neq i$, and of the intersection of $A(s)$ with the boundary face $S^n(I^-(s)) = \{x \in S^n \mid x_k = 0,\ k \in I^-(s)\}$ of S^n. Next, for nonempty proper subsets K of I_{n+1}, let $p(K)$ be the relative projection of v on the

boundary face $S^n(I_{n+1}/K)$, i.e.,

$$p_i(K) = v_i \Big/ \left(\sum_{k \in K} v_k \right), \quad i \in K,$$

$$p_i(K) = 0, \qquad i \notin K.$$

When the sign vector s contains no zeroes, the set $A(s)$ is the line segment connecting v and its projection $p(I^+(s))$ on $S^n(I^-(s))$. These $2^{n+1}-2$ one-dimensional sets $A(s)$ are the rays of the algorithm. As described in Section 2, the algorithm leaves v along the ray $A(s^0)$ for which $s^0 = \operatorname{sgn} z(v)$, where as in the sequel sgn is taken componentswise. In general, a point x in S^n satisfies (2.1) if and only if for some sign vector s, x lies in $A(s)$ and $s = \operatorname{sgn} Z(x)$, so that the algorithm in fact traces from v the path of points x in S^n for which x lies in $A(\operatorname{sgn} Z(x))$.

The triangulation of S^n with respect to which the piecewise linear approximation Z of z is defined must be such that it triangulates each $A(s)$. The V-triangulation of S^n introduced in Doup and Talman [1] satisfies this property. First, each (t-dimensional) set $A(s)$ is subdivided in t-dimensional subsets $A(s, \gamma(s))$ with $\gamma(s) = (k_1, \ldots, k_{t-1})$ any permutation of the $t-1$ elements of $I^0(s)$.

Definition 3.2. The set $A(s, \gamma(s))$ is given by

$$A(s, \gamma(s)) = \Big\{ x \in S^n \,\Big|\, x = v + \beta q(0) + \sum_i \alpha(k_i) q(k_i),$$

$$\text{with } 0 \leqslant \alpha(k_{t-1}) \leqslant \cdots \leqslant \alpha(k_1) \leqslant \beta \leqslant 1 \Big\}, \tag{3.1}$$

where the $(n+1)$-vector $q(0)$ is given by

$$q(0) = p(I^+(s)) - v,$$

and where for $i = 1, \ldots, t-1$ the $(n+1)$-vector $q(k_i)$ is given by

$$q(k_i) = p(I^+(s) \cup \{k_1, \ldots, k_i\}) - p(I^+(s) \cup \{k_1, \ldots, k_{i-1}\}).$$

Note that the rank of the $(n+1) \times t$ matrix $Q(s, \gamma(s))$ with first column $q(0)$ and $(i+1)$th column $q(k_i)$, $i = 1, \ldots, t-1$, is equal to $t = |I^0(s)| + 1$, so that the dimension of $A(s, \gamma(s))$ is indeed t. $A(s)$ is the union of $A(s, \gamma(s))$ over all permutations $\gamma(s)$. Some sets are illustrated in Figure 1 for $n = 3$. The boundary of $A(s, \gamma(s))$ is the union of $(t-1)$-dimensional subsets, each of them obtained by setting exactly one inequality in (3.1) to an equality. If $\beta = 1$ we obtain the intersection of $A(s, \gamma(s))$ with $S^n(I^-(s))$.

The V-triangulation of S^n is the union of the triangulations of the n-dimensional subsets $A(s, \gamma(s))$ into n-simplices, each of which is the K-triangulation of R^n restricted to the set $\{x \in R^n \,|\, 0 \leqslant x_n \leqslant x_{n-1} \leqslant \cdots \leqslant x_1 \leqslant 1\}$ and transformed by the affine mapping $Q(s, \gamma(s))x + v$. This triangulation induces a subdivision of each

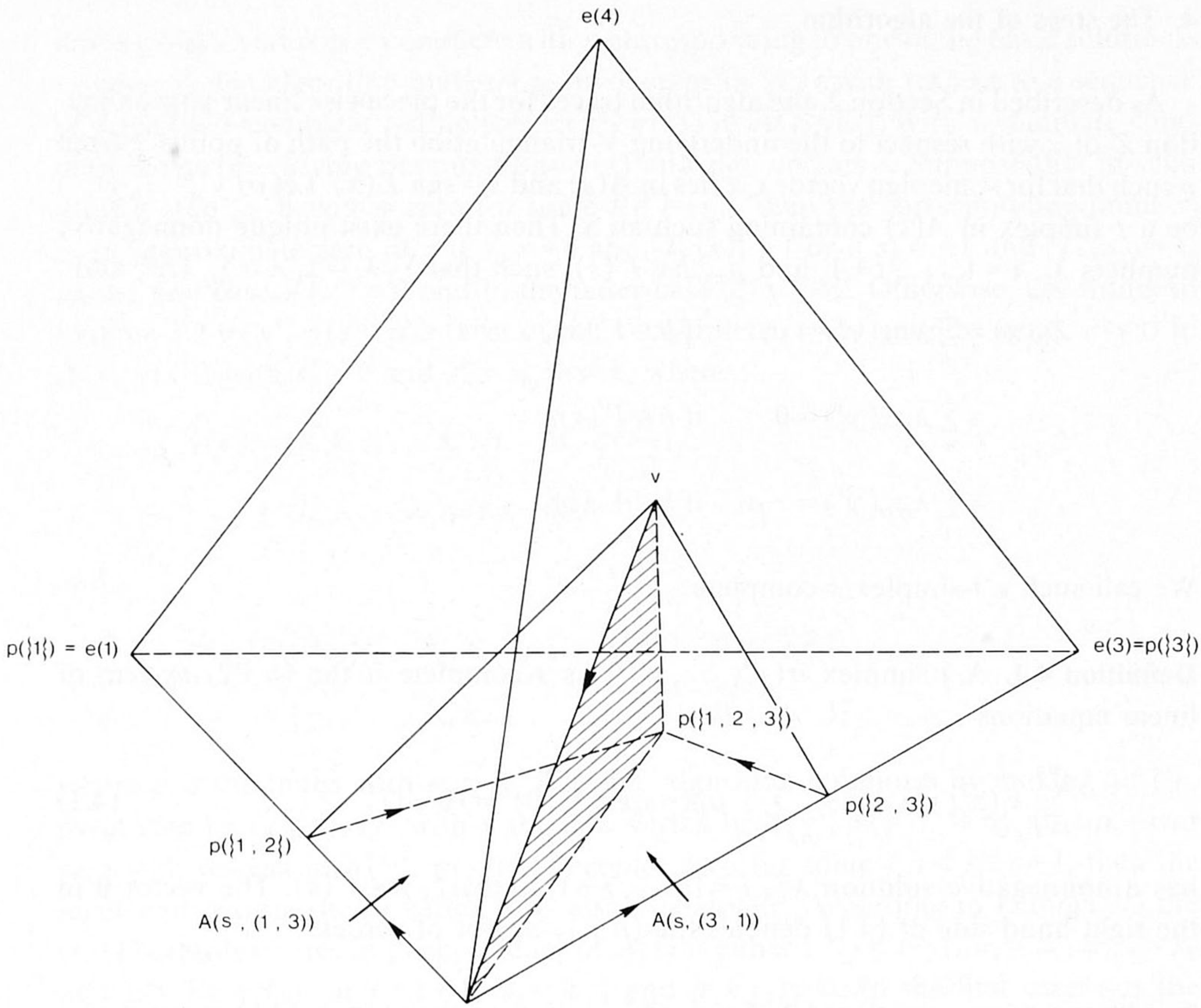

Fig. 1. Illustration of $A(s)$, $s=(0,+1,\ 0,\ -1)^{\mathrm{T}}$, which is subdivided into $A(s,(1,3))$ and $A(s,(3,1))$; $\dim A(s)=|I^0(s)|+1=3$.

t-dimensional set $A(s,\gamma(s))$ into t-simplices. For simplicity in the sequel we set $k_0=0$.

Lemma 3.3 *Let S^n be triangulated according to the V-triangulation with grid size m^{-1}. Then $A(s,\gamma(s))$ with $\gamma(s)=(k_1,\ldots,k_{t-1})$ is triangulated by the collection $G(s,\gamma(s))$ of t-simplices $\sigma(y^1,\pi(s))$ with vertices $y^1,\ldots,y^{t+1}$ such that*

(i) $y^1=v+\sum_i a(k_i)m^{-1}q(k_i)$ *for nonnegative integers* $a(k_i)$, $i=0,\ldots,t-1$, *with* $0\leqslant a(k_{t-1})\leqslant\cdots\leqslant a(k_0)\leqslant m-1$

(ii) $\pi(s)=(\pi_1,\ldots,\pi_t)$ *is a permutation of the t elements* $k_0,\ldots,k_{t-1}$ *such that* $p>p'$ *if for some i,* $1\leqslant i\leqslant t-1$, $\pi_p=k_i$, $\pi_{p'}=k_{i-1}$ *and* $a(k_i)=a(k_{i-1})$

(iii) $y^{i+1}=y^i+m^{-1}q(\pi_i)$, $i=1,\ldots,t$.

The union $G(s)$ of the collections $G(s,\gamma(s))$ over all permutations $\gamma(s)$ triangulates $A(s)$.

4. The steps of the algorithm

As described in Section 2, the algorithm traces for the piecewise linear approximation Z of z with respect to the underlying V-triangulation the path of points x from v such that for some sign vector s, x lies in $A(s)$ and $s = \operatorname{sgn} Z(x)$. Let $\sigma(y^1, \ldots, y^{t+1})$ be a t-simplex in $A(s)$ containing such an x. Then there exist unique nonnegative numbers λ_i, $i = 1, \ldots, t+1$, and μ_h, $h \notin I^0(s)$, such that $\sum_i \lambda_i = 1$, $x = \sum_i \lambda_i y^i$, and

$$\begin{aligned} Z_h(x) &= \sum_i \lambda_i z_h(y^i) = \mu_h \qquad \text{if } h \in I^+(s), \\ &= \sum_i \lambda_i z_h(y^i) = 0 \qquad \text{if } h \in I^0(s), \\ &= \sum_i \lambda_i z_h(y^i) = -\mu_h \quad \text{if } h \in I^-(s). \end{aligned}$$

We call such a t-simplex s-complete.

Definition 4.1. A t-simplex $\sigma(y^1, \ldots, y^{t+1})$ is s-complete if the $(n+2)$-system of linear equations

$$\sum_{i=1}^{t+1} \lambda_i (z^{\mathrm{T}}(y^i), 1)^{\mathrm{T}} + \sum_{h \notin I^0(s)} \mu_h (-s_h e^{\mathrm{T}}(h), 0)^{\mathrm{T}} = (\mathbf{0}^{\mathrm{T}}, 1)^{\mathrm{T}} \tag{4.1}$$

has a nonnegative solution λ_i^*, $i = 1, \ldots, t+1$, and μ_h^*, $h \notin I^0(s)$. The vector $\mathbf{0}$ in the right hand side of (4.1) denotes the $(n+1)$-vector of zeroes.

Nondegeneracy assumption 4.2. For each solution to the system (4.1) at most one of the variables λ_i and μ_h is equal to zero.

Under this assumption the system (4.1) has a line segment of solutions (λ^*, μ^*), if any. An end point of such a line segment is called a basic solution and has exactly one of the variables equal to zero. The line segment of solutions (λ, μ) induces a line segment of points $x = \sum_i \lambda_i y^i$ in σ for which according to (4.1) $\operatorname{sgn} Z(x) = \operatorname{sgn}(\sum_i \lambda_i z(y^i)) = s$. When σ lies in $A(s)$ such an x satisfies (2.1). At an end point x' of a line segment either $\mu_h^* = 0$ in (4.1) and hence, in (2.1), $Z_h(x') = 0$ for some $h \notin I^0(s)$, or for some i $\lambda_i^* = 0$ in (4.1) and hence x' lies in the facet of σ opposite the vertex y^i. The path of points satisfying (2.1) can therefore be followed by making linear programming pivot steps in the system (4.1) for a sequence of s-complete adjacent t-simplices $\sigma(y^1, \pi(s))$ in $A(s)$ for varying sign vectors s. Now, with $s^0 = \operatorname{sgn} z(v)$, let $\sigma^0(v, (0))$ be the unique one-simplex in $A(s^0)$ having v as a vertex. Then σ^0 is s^0-complete with $\lambda_2 = 0$ at one of the basic solutions. The algorithm starts by making a linear programming pivot step with $(z^{\mathrm{T}}(y^2), 1)^{\mathrm{T}}$ in the corresponding system of linear equations (4.1) where $y^2 = v + m^{-1} q(0)$. Notice that, because of the nondegeneracy assumption, s^0 does not contain any zero and hence there is no other sign vector s with $|I^0(s)| = 0$ for which the unique one-simplex in $A(s)$

having v as a vertex is s-complete with v corresponding to one of the basic solutions. In general, the algorithm makes 1.p. pivot steps in (4.1) with respect to a sequence of adjacent s-complete t-simplices $\sigma(y^1, \pi(s))$ in $A(s, \gamma(s))$ with s-complete common facets for varying permutations $\gamma(s)$ and sign vectors s. Suppose that in such a pivot step μ_k becomes zero for some $k \notin I^0(s)$. Then the corresponding point x' is an approximate zero of z if $s_k = +1$ and $|I^+(s)| = 1$ or if $s_k = -1$ and $|I^-(s)| = 1$. In the first case $Z(x') \leqslant 0$ and in the latter case $Z(x') \geqslant 0$. Otherwise, according to Lemma 3.3 $\sigma(y^1, \pi(s))$ is a facet of the s'-complete $(t+1)$-simplex $\sigma(y^1, \pi(s'))$ in $A(s', \gamma(s'))$ with $s'_k = 0$ and $s'_h = s_h$, $h \neq k$, where

$$\gamma(s') = (k, k_1, \ldots, k_{t-1}) \quad \text{if } s_k = +1,$$

$$= (k_1, \ldots, k_{t-1}, k) \quad \text{if } s_k = -1,$$

and

$$\pi(s') = (\pi_1, \ldots, \pi_p, k, \pi_{p+1}, \ldots, \pi_t) \quad \text{if } s_k = +1$$

$$= (\pi_1, \ldots, \pi_t, k) \quad \text{if } s_k = -1,$$

where p is the index with $\pi_p = 0$. Then the algorithm continues by making an 1.p. pivot step by $(z^{\mathrm{T}}(y), 1)^{\mathrm{T}}$ with y the new vertex of $\sigma(y^1, \pi(s'))$. If by an 1.p. pivot step with respect to $\sigma(y^1, \pi(s))$ λ_j becomes zero for some j, $1 \leqslant j \leqslant t+1$, then the facet τ of σ opposite to vertex y^j is also s-complete. According to Lemma 3.3 the $(t-1)$-simplex τ lies in the boundary of $A(s)$ if either $1 < j < t+1$, $\pi_j = k_1$, $\pi_{j-1} = k_0$ and $a(k_1) = a(k_0)$, or $j = t+1$, $\pi_t = k_{t-1}$ and $a(k_{t-1}) = 0$. In the first case τ is the s'-complete $(t-1)$-simplex $\sigma(y^1, \pi(s'))$ in $A(s', \gamma(s'))$ with $s'_k = +1$ for $k = k_1$ and $s'_h = s_h$, $h \neq k$, where

$$\gamma(s') = (k_2, \ldots, k_{t-1})$$

and

$$\pi(s') = (\pi_1, \ldots, \pi_{j-1}, \pi_{j+1}, \ldots, \pi_t).$$

In the other case τ is the s'-complete $(t-1)$-simplex $\sigma(y^1, \pi(s'))$ in $A(s', \gamma(s'))$ with $s'_k = -1$ for $k = k_{t-1}$ and $s'_h = s_h$, $h \neq k$, where

$$\gamma(s') = (k_1, \ldots, k_{t-2})$$

and

$$\pi(s') = (\pi_1, \ldots, \pi_{t-1}).$$

In both cases the algorithm continues by making an 1.p. pivot step in (4.1) with $(-s_k e^{\mathrm{T}}(k), 0)^{\mathrm{T}}$. Recall that an s-complete facet τ of a t-simplex $\sigma(y^1, \ldots, y^{t+1})$ in $A(s)$ cannot lie in the boundary $S^n(I^-(s))$, since $p_i = 0$ implies $Z_i(p) = \sum_h \lambda_h z_i(y^h) > 0$ for all p in τ.

Finally we have to consider the case that if in (4.1) λ_j becomes zero for some j, $1 \leqslant j \leqslant t+1$, the facet τ opposite the vertex y^j of σ does not lie in the boundary of $A(s)$. Then τ is a facet of just one other t-simplex σ' in $A(s)$. More precisely, if for some i, $2 \leqslant i \leqslant t-1$, $\pi_j = k_i$, $\pi_{j-1} = k_{i-1}$ and $a(k_i) = a(k_{i-1})$, then according to Definition 3.2 and Lemma 3.3 σ' is the t-simplex $\sigma(y^1, \pi'(s))$ in $A(s, \gamma'(s))$ with

$$\gamma'(s) = (k_1, \ldots, k_{i-2}, k_i, k_{i-1}, k_{i+1}, \ldots, k_{t-1})$$

and

$$\pi'(s) = (\pi_1, \ldots, \pi_{j-2}, \pi_j, \pi_{j-1}, \pi_{j+1}, \ldots, \pi_t).$$

Otherwise σ' is the t-simplex $\sigma(y'^1, \pi'(s))$ in $A(s, \gamma(s))$ obtained from σ according to the standard replacement rule of the K-triangulation in R^n, as described in Table 1, where $a_h = a(0)$, $h \in I^+(s)$, $a_h = a(h)$, $h \in I^0(s)$, $a_h = 0$, $h \in I^-(s)$, and where $e(0)$ is the $(n+1)$-vector given by $e_i(0) = 1$ when $i \in I^+(s)$ and $e_i(0) = 0$ otherwise. In each of the latter cases the algorithm continues by making an 1.p. pivot step in (4.1) with $(z^T(y), 1)^T$, where y is the new vertex of σ' opposite the facet shared with σ.

Table 1
j is the index of the vertex of $\sigma(y^1, \pi(s))$ to be replaced

	y^1 becomes	$\pi(s)$ becomes	a becomes
$j=1$	$y^1 + m^{-1}q(\pi_1)$	$(\pi_2, \ldots, \pi_t, \pi_1)$	$a + e(\pi_1)$
$1<j<t+1$	y^1	$(\pi_1, \ldots, \pi_j, \quad \pi_{j-1}, \ldots, \pi_t)$	a
$j=t+1$	$y^1 - m^{-1}q(\pi_t)$	$(\pi_t, \pi_1, \ldots, \pi_{t-1})$	$a - e(\pi_t)$

In this way the algorithm traces the piecewise linear path of points $x = \sum_i \lambda_i y^i$ from v satisfying (2.1) and followed by a unique sequence of adjacent simplices of varying dimension. Under the nondegeneracy assumption no simplex can be generated more than once. Since the number of simplices of the underlying triangulation is finite, the algorithm must terminate within a finite number of steps with a point x^* for which either $Z(x^*) \leqslant 0$ or $Z(x^*) \geqslant 0$. If the accuracy of this approximate solution is not satisfactory the algorithm can be restarted in x^* with a finer triangulation. More generally, the set of s-complete simplices $\sigma(y^1, \pi(s))$ in $A(s)$ for all sign vectors s for which both $|I^-(s)|$ and $|I^+(s)|$ are positive form sequences of adjacent simplices with s-complete facets in common. Each such sequence which is not a loop traces a path of points satisfying (2.1) and has two end points. Exactly one path connects the point v with an approximating equilibrium and is followed by the algorithm. All other paths with two end points connect two different approximating equilbria.

5. Computational results

The algorithm presented in this paper has been applied to the three pure exchange economies given in Scarf [13] and to a pure exchange economy with fifteen commodities and five consumers. More precisely, for the demand function of commodity j

$$d_j(p) = \sum_{h=1}^{5} \left(a_{hj} p_j^{-b_h} \sum_{k=1}^{15} w_{hk} p_k \Big/ \sum_{k=1}^{15} a_{hk} p_k^{1-b_h} \right), \quad j = 1, \ldots, 15,$$

the numbers a_{hj} and w_{hj}, $j = 1, \ldots, 10$, and b_h for all h, are the same as for the third economy and the remaining elements a_{hj} and w_{hj}, $h = 1, \ldots, 5$, $j = 11, \ldots, 15$ are given in Tables 2 and 3 respectively.

For all the four examples the new algorithm has been compared with the $(n+1)$-ray algorithm as described in Doup and Talman [1]. In both algorithms the V-triangulation is the underlying simplicial subdivision of S^n. In all runs for both algorithms the initial starting point is the barycenter of S^n and the initial grid size is equal to $\frac{1}{2}$. When an approximating equilibrium has been found the algorithm is restarted in that point with a smaller grid size. The grid is refined with a factor of two. The grid refinement is stopped when the excess demands at the approximate solution are less than 10^{-8}. The results for the two different algorithms are given in Table 4, where for each economy two cases are discussed. Case one allows quasi-Newton steps to be performed and case two only allows restarts of the variable

Table 2

The elements a_{hj}, $h = 1, \ldots, 5$, $j = 11, \ldots, 15$

h \ j	11	12	13	14	15
1	2.5	0.8	1.4	4.0	3.6
2	1.0	1.0	1.0	1.0	1.0
3	2.3	4.5	3.0	0.9	7.9
4	11.0	12.0	13.0	14.0	15.0
5	3.0	6.0	0.8	7.0	12.0

Table 3

The elements a_{hj}, $h = 1, \ldots, 5$, $j = 11, \ldots, 15$

h \ j	11	12	13	14	15
1	7.9	3.1	5.3	4.0	2.0
2	8.0	7.0	6.0	5.0	4.0
3	10.0	3.0	7.0	5.0	1.5
4	6.0	4.6	2.0	11.0	0.4
5	4.8	6.1	3.2	9.4	0.9

Table 4

The number of function evaluations and the number of quasi-Newton steps for the two algorithms

		$(n+1)$-ray algorithm	$(2^{n+1}-2)$-ray algorithm
Economy 1	1	30 (9)	31 (5)
	2	45	42
Economy 2	1	63 (5)	39 (7)
	2	78	72
Economy 3	1	72 (7)	69 (5)
	2	132	88
Economy 4	1	119 (4)	131 (6)
	2	277	192

dimension algorithm. The number in brackets denotes the number of Quasi-Newton steps.

References

[1] T.M. Doup and A.J.J. Talman, "A new simplicial variable dimension algorithm to find equilibria on the product space of unit simplices," *Mathematical Programming* 37 (1987) 319–355.

[2] M. Kojima and Y. Yamamoto, "Variable dimension algorithms: Basic theory, interpretations and extension of some existing methods," *Mathematical Programming* 24 (1982) 177–215.

[3] H.W. Kuhn, "Simplicial approximation of fixed points," *Proceedings of the National Academy of Science* 61 (1968) 1238–1242.

[4] H.W. Kuhn, "Approximate search for fixed points," *Computing Methods in Optimization Problems* 2 (1969) 199–211.

[5] H.W. Kuhn and J.G. MacKinnon, "The Sandwich method for finding fixed points," *Journal of Optimization Theory and Applications* 17 (1975) 189–204.

[6] G. van der Laan and A.J.J. Talman, "A restart algorithm for computing fixed points without an extra dimension," *Mathematical Programming* 17 (1979) 74–84.

[7] G. van der Laan and A.J.J. Talman, "An improvement of fixed point algorithms by using a good triangulation," *Mathematical Programming* 18 (1980) 274–285.

[8] G. van der Laan and A.J.J. Talman, "Note on the path following approach of equilibrium programming," *Mathematical Programming* 25 (1983) 363–367.

[9] G. van der Laan and A.J.J. Talman, "Adjustment processes for finding economic equilibria," Onderzoeksverslag 141, Free University, Amsterdam, 1985.

[10] D.G. Saari, "Iterative price mechanisms," *Econometrica* 53 (1985) 1117–1131.

[11] H. Scarf, "Some examples of global instability of the competitive equilibrium," *International Economic Review* 1 (1960), 157–172.

[12] H. Scarf, "The approximation of fixed points of a continuous mapping," *SIAM Journal of Applied Mathematics* 15 (1967) 1328–1343.

[13] H. Scarf, *The Computation of Economic Equilibria* (Yale University Press, New Haven, CT, 1973).

[14] W.I. Zangwill and C.B. Garcia, "Equilibrium programming: the path following approach and dynamics," *Mathematical Programming* 21 (1981) 262–289.